■ อรรคชัย กัมลาศพิทักษ์

Proxy/Cache Server

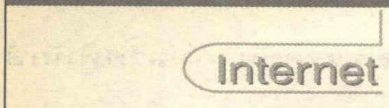
รู้จักกับ Proxy และการติดตั้งโปรแกรม Apache ให้ทำงานเป็น Proxy

ในการใช้บริการอินเทอร์เน็ตด้วยโปรโตคอลอย่าง HTTP, FTP, Gopher และ WAIS ข้อมูลต่าง ๆ จะได้รับการจัดเตรียมไว้ล่วงหน้าแล้ว หากไม่มีการเปลี่ยนแปลงข้อมูลที่เซิร์ฟเวอร์ การเข้าไปขอใช้บริการในแต่ละครั้งก็มักจะได้รับข้อมูลเหมือนเดิมทุกประการ จะมีข้อยกเว้นในบางกรณีเท่านั้น จะเห็นว่าหากการเรียกดูข้อมูลที่ซ้ำ ๆ กันนี้แล้วเราต้องโหลดข้อมูลจากเซิร์ฟเวอร์ปลายทางมาใหม่ทุกครั้ง จึงเป็นการสิ้นเปลืองทรัพยากรต่าง ๆ เพิ่มขึ้น และเพิ่มความคับคั่งของวงจรบนเครือข่าย โดยไม่จำเป็น เว็บเบราเซอร์จึงได้จัดพื้นที่ส่วนหนึ่งบนฮาร์ดดิสก์สำหรับเก็บข้อมูลที่เคยเรียกดูจากเซิร์ฟเวอร์แล้ว หากย้อนกลับไปดูใหม่ก็สามารถนำข้อมูลจากแคชนี้ออกมาใช้งานได้ทันที โดยไม่จำเป็นต้องไปขอข้อมูลจากเซิร์ฟเวอร์ปลายทางมาอีก ทำให้การเรียกดูข้อมูลนี้ทำได้เร็วขึ้น

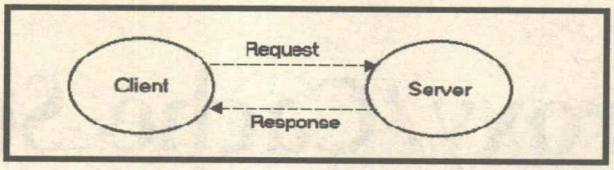
หากมองในระดับที่กว้างขึ้น อย่างเช่นในหน่วยงานที่มีผู้ใช้งานร่วมกันหลายคน หรือผู้ให้บริการอินเทอร์เน็ตซึ่งมีผู้ใช้บริการจำนวนมาก ข้อมูลต่าง ๆ ที่ใช้งานก็มักจะมีบางส่วนที่ซ้ำ ๆ กัน หากสามารถใช้ข้อมูลเหล่านี้ร่วมกันได้แล้วก็จะลดความคับคั่งของวงจรสื่อสารลงได้ อีกทั้งยังทำให้ผู้ใช้บริการรายหลังที่มีข้อมูลอยู่แล้วสามารถเรียกดูด้วยความรวดเร็วกว่าอีกด้วย การแชร์หรือใช้ข้อมูลร่วมกันนี้สามารถทำได้ด้วยการนำ Proxy/Cache Server มาใช้งาน แต่นั่นเป็นเพียงประโยชน์ส่วนหนึ่งของ Proxy เท่านั้น บทความนี้จะแนะนำให้รู้จักกับ Proxy และการติดตั้งโปรแกรม Apache ให้ทำงานเป็น Proxy

Proxy Server คืออะไร

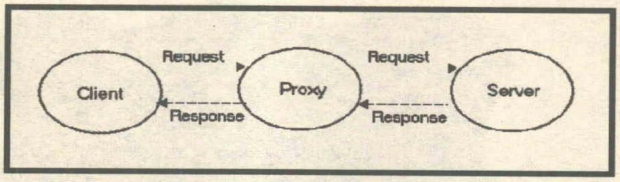
เรามักจะคุ้นเคยกับคำว่า Proxy คู่กับคำว่า Server ทั้งนี้เนื่องจากในมุมมองของผู้ใช้อินเทอร์เน็ตทั่วไป Proxy ทำหน้าที่เป็นเซิร์ฟเวอร์ที่คอยตอบสนองความต้องการของผู้ใช้งาน ไม่ว่าจะต้องการเข้าไปยังเว็บไซท์ใด Proxy จะทำหน้าที่นำข้อมูลหรือเอกสารที่ต้องการมาให้เสมอ แต่ในสภาพการทำงานที่แท้จริงแล้ว Proxy ทำหน้าที่เป็นตัวกลางระหว่างไคลเอนต์และเซิร์ฟเวอร์ ในขณะที่รับคำร้องขอจากไคลเอนต์ Proxy จะทำหน้าที่เป็นเซิร์ฟเวอร์ แต่เมื่อได้รับคำร้องขอนั้น ๆ แล้ว Proxy จะติดต่อกับเซิร์ฟเวอร์ปลายทางเพื่อร้องขอสิ่งที่ไคลเอนต์ต้องการ ในช่วงนี้ Proxy จะทำหน้าที่เป็นไคลเอนต์แทน เมื่อได้รับข้อมูลหรือเอกสารจากเซิร์ฟเวอร์ปลายทางแล้ว Proxy จะทำหน้าที่ส่งต่อเอกสารเหล่านั้นกลับไปยังไคลเอนต์ ในช่วงนี้ Proxy จะทำหน้าที่เป็นเซิร์ฟเวอร์อีกครั้งหนึ่ง Proxy จึงเป็นลูกผสมระหว่างไคลเอนต์และเซิร์ฟเวอร์ในตัวเดียวกัน



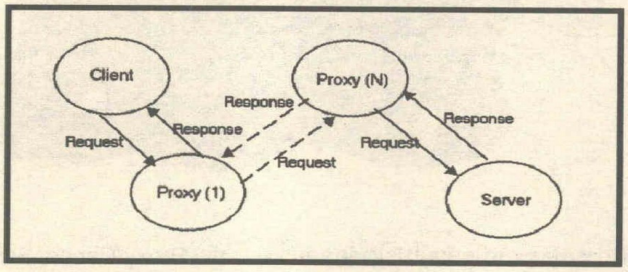
รูปที่ 1 ➲ การติดต่อโดยตรงระหว่างไคลเอนต์เซิร์ฟเวอร์

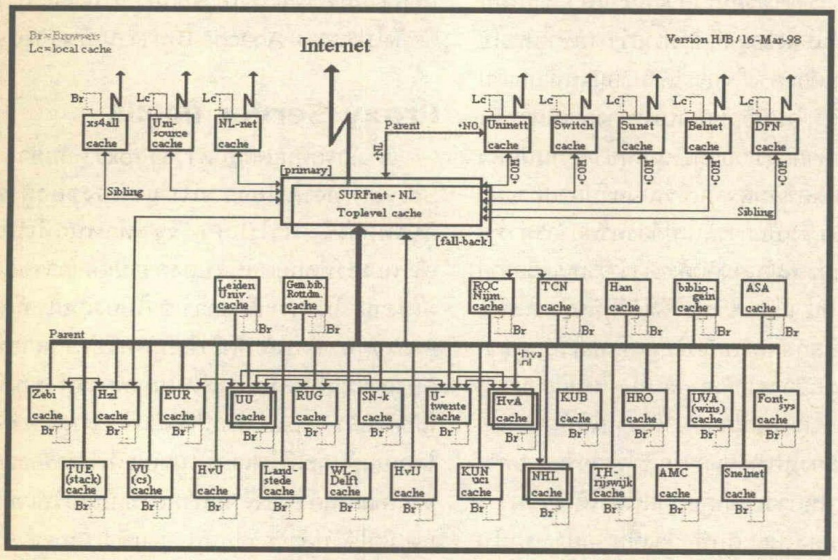


รูปที่ 2 ➲ การติดต่อระหว่างไคลเอนต์เซิร์ฟเวอร์ผ่าน Proxy



รูปที่ 3 การติดต่อระหว่างไคลเอนต์เซิร์ฟเวอร์กรณีผ่าน Proxy หลาย ๆ เครื่อง





รูปที่ 4 การเชื่อมโยงของ Proxy/Cache Server ของ SURFnet ผู้ให้บริการอินเทอร์เน็ตของเนเธอร์แลนด์

โดยทั่วไป Proxy มักจะทำการเก็บข้อมูลที่เคยโหลดมาจากเซิร์ฟเวอร์ปลายทางแล้วเก็บไว้ในแคช จึงมักจะเรียกว่า Proxy/Cache Server หรือบางครั้งอาจจะเรียกว่า Cache Server โดยไม่มีคำว่า Proxy ก็ได้

ในสภาพของการทำงานจริง Proxy แต่ละเครื่องอาจต่อเข้ากับไคลเอนต์มากกว่าหนึ่งเครื่อง และ Proxy ก็สามารถเชื่อมโยงกับเซิร์ฟเวอร์ปลายทางมากกว่าหนึ่งเครื่องด้วยเช่นเดียวกัน รวมถึง Proxy ที่เชื่อมโยงกับ Proxy เครื่องอื่น ๆ ด้วยอาจให้บริการกับไคลเอนต์และ Proxy อื่นในเวลาเดียวกัน รูปที่ 2 และ 3 ควรจะสลับซับซ้อนกว่าที่แสดงให้เห็นมาก โดยเฉพาะรูปที่ 3 การเชื่อมโยงระหว่าง Proxy แต่ละเครื่องในเครือข่ายมักจะเชื่อมโยงกันเป็นตาข่ายที่เรียกว่า Mesh โดยอาจมี Proxy หลักเพียงหนึ่งเครื่อง ส่วนที่เหลือจะใช้ข้อมูลจาก Proxy หลักนี้ ซึ่งทำให้สามารถประหยัดค่าใช้จ่ายด้านวงจรสื่อสารลงได้มากจากการใช้ข้อมูลร่วมกัน วิธีการนี้มักใช้โดยผู้ให้บริการอินเทอร์เน็ตหรือ ISP รูปที่ 4 แสดงให้เห็นถึงการเชื่อมโยง Proxy ต่าง ๆ บนเครือข่ายของ SURFnet ซึ่งเป็นผู้ให้บริการอินเทอร์เน็ตในเนเธอร์แลนด์ จะเห็นว่ามีการเชื่อมโยง Proxy อยู่ 2 แบบคือ แบบพ่อ/แม่-ลูก (Parent-Child) และแบบพี่-น้อง (Sibling) ข้อแตกต่างระหว่างทั้ง 2 แบบก็คือ กรณีเชื่อมโยงแบบ Parent-Child เครื่องที่ทำหน้าที่เป็น Child จะต้องพึ่งพิงเครื่อง Parent อย่างมาก โดยเครื่อง Parent จะต้องรับผิดชอบในการหาข้อมูลที่เครื่อง Child ต้องการมาให้ แม้ไม่มีข้อมูลดังกล่าวในแคช ก็จะติดต่อกับเซิร์ฟเวอร์ปลายทางหรือ Proxy อื่นเพื่อนำข้อมูลดังกล่าวมาให้ แต่กรณีของการเชื่อมโยงแบบ Sibling ความช่วยเหลือจะจำกัดอยู่เพียงเท่าที่พอจะให้ข้อมูลได้ โดยเครื่อง Sibling จะช่วยเหลือในการจัดส่งข้อมูลให้เฉพาะที่มีอยู่ในแคชเท่านั้น หากข้อมูลที่ต้องการไม่มีอยู่ในแคช Sibling จะไม่รับผิดชอบในการติดต่อขอข้อมูลนั้น ๆ มาให้อีก เราต้องหาวิธีติดต่อขอข้อมูลจากทางอื่นแทน จะเห็นว่าเครื่องที่เป็น Parent จะมีภาระที่หนักกว่าเครื่องที่เป็น Sibling

การประยุกต์ใช้งาน Proxy

มีเหตุผลอยู่หลายประการในการนำ Proxy/Cache มาใช้งาน ดังนี้

1. เครื่องที่คุณใช้งานอยู่หลัง Firewall
2. คุณต้องการความเร็วในการใช้อินเทอร์เน็ต
3. คุณมี IP Address สำหรับอินเทอร์เน็ตไม่เพียงพอกับจำนวนเครื่องที่ใช้งาน

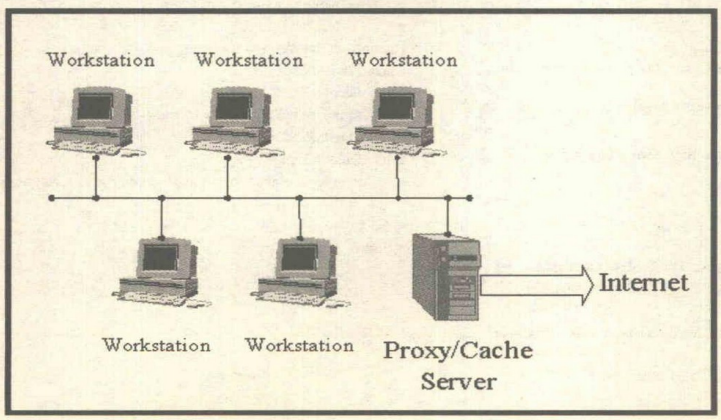
Proxy ถูกนำมาประยุกต์ใช้งานในหลายลักษณะ แบบที่ง่ายที่สุดก็คือในรูปที่ 2 ซึ่งใช้เป็นตัวกลางในการติดต่อเข้าสู่อินเทอร์เน็ต บางกรณี Proxy อาจถูกใช้เป็น Firewall หรือ Gateway ระหว่างเครือข่ายมากกว่าหนึ่งเครือข่าย

ผู้ใช้บริการอินเทอร์เน็ตส่วนใหญ่มักจะเข้าไปใช้บริการจากเว็บไซท์ที่ซ้ำ ๆ กัน ดังนั้นข้อมูลที่ถูกเรียกดูจึงมักจะซ้ำ ๆ กันด้วย ข้อมูลที่เคยโหลดมาแล้ว หากเก็บไว้ในหน่วยความจำแคช เมื่อผู้ใช้งานคนต่อไปมาเรียกดูอีกก็สามารถนำจากแคชมาใช้งานได้โดยไม่ต้องไปโหลดมาจากเซิร์ฟเวอร์อีก ความคับคั่งของการใช้งานวงจรจะลดลงไปด้วย ยิ่งมีผู้เรียกดูข้อมูลเดียวกันซ้ำ ๆ มากเท่าไรก็จะให้ความรู้สึกว่าวงจรที่เคยใช้งานอยู่นี้สามารถรองรับการใช้งานของผู้ใช้งานจำนวนที่มากขึ้นโดยไม่จำเป็นต้องขยายความจุของวงจรที่ติดต่อกับเครือข่ายอินเทอร์เน็ต ช่วยลดต้นทุนขององค์กรนั้น ๆ ลงด้วย นอกจากนี้การใช้งาน Proxy โดยปกติจะทำให้ผู้ใช้มีความรู้สึกว่าติดต่อกับอินเทอร์เน็ตด้วยความเร็วที่สูงขึ้น เนื่องจากการโหลดข้อมูลจาก Proxy กรณีที่มีข้อมูลอยู่ในแคชแล้ว สามารถทำได้รวดเร็วกว่าการโหลดจากเซิร์ฟเวอร์ปลายทางโดยตรง เกือบเท่ากับความเร็วสูงสุดที่เราใช้ในการติดต่อกับเซิร์ฟเวอร์ของเรา โดยเฉพาะเมื่อใช้งานบน LAN จะให้ความรู้สึกว่าเร็วมาก

จัดหา Proxy/Cache Server มาใช้งาน

ปัจจุบันการจัดทำ Proxy เพื่อใช้งานในหน่วยงานต่าง ๆ เป็นเรื่องที่ทำได้ไม่ยาก เนื่องจากมีโปรแกรมที่ทำหน้าที่นี้จำนวนมากที่แจกจ่ายให้ใช้งานกันในเกือบทุกระบบปฏิบัติการ ไม่ว่าจะเป็น Unix/Linux, Windows, Macintosh หรืออื่น ๆ โปรแกรมที่ได้รับความนิยมได้แก่ CERN, SQUID, Apache, Netscape Proxy Server, WinProxy, Wingate, CSM Proxy, Microsoft Proxy Server เป็นต้น ส่วนใหญ่แจกจ่ายให้ใช้งานฟรี โดยเฉพาะบนระบบปฏิบัติการ Unix/Linux

รูปที่ 5 การใช้ Proxy/Cache Server ในการเชื่อมโยงสู่อินเทอร์เน็ต



การติดตั้ง Apache เป็น Proxy Server สำหรับระบบ Unix

สำหรับระบบปฏิบัติการ Unix/Linux ถึงแม้ว่าโปรแกรม SQUID จะได้รับความนิยมในการใช้งานเป็น Proxy/Cache Server เป็นอย่างมากก็ตาม แต่ Apache ซึ่งเป็นโปรแกรมที่ได้รับความนิยมสูงที่สุดในการทำงานเป็น httpd หรือเว็บเซิร์ฟเวอร์ ก็มีความสามารถในการทำงานเป็น Proxy/Cache Server ได้ไม่ด้อยไปกว่ากัน Apache เป็นโปรแกรมที่ได้รับการพัฒนามายาวนาน มีความยืดหยุ่นในการใช้งานสูง และมีให้ใช้งานในทั้งระบบ Unix/Linux และ Windows ทำให้มีผู้ใช้งาน Apache มากกว่าครึ่งหนึ่งของเว็บเซิร์ฟเวอร์ทั้งหมดทั่วโลก และมีแนวโน้มว่าสัดส่วนการใช้งานนี้จะเพิ่มขึ้นเรื่อย ๆ ถ้าคุณใช้ Apache เป็นเว็บเซิร์ฟเวอร์สำหรับอินเทอร์เน็ตหรืออินทราเน็ตอยู่ คุณก็สามารถกำหนดให้ทำงานเป็น Proxy/Cache Server ได้ด้วย โปรแกรมนี้สามารถดาวน์โหลดมาใช้งานได้ฟรีจาก http://www.apache.org

การติดตั้ง Apache เพื่อให้ทำหน้าที่เป็น Proxy จะต้องเตรียมการตั้งแต่ช่วงของการคอมไพล์โปรแกรม ในกรณีของเครื่องที่ใช้บนระบบปฏิบัติการ Unix/Linux เมื่อดาวน์โหลดโปรแกรมมาแล้ว ให้ทำการแตกไฟล์ออกด้วยคำสั่ง

```
#gzip -d   apache_X.X.X.tar.gz
#tar -xvf   apache_X.X.X.tar
```

โดยที่ X.X.X เป็นเวอร์ชันของโปรแกรมที่ต้องการใช้งาน ซอร์สโค้ดของโปรแกรมที่ได้จะอยู่ในไดเรคทอรีย่อย src/ เราจะต้องแก้ไขในไฟล์ Configuration ให้ทำการรวมส่วนที่ทำหน้าที่เป็น Proxy เข้ากับโปรแกรมที่ต้องการด้วย โดยการคอมไพล์รวมส่วน Module Proxy ไว้ด้วย เนื่องจากการคอมไพล์โปรแกรมโดยปกติจะไม่รวมส่วนนี้เข้าไปด้วย วิธีการก็คือใช้ Text Editor เปิดไฟล์ชื่อ Configuration ค้นหาบรรทัดที่มีคำว่า # AddModule modules/proxy/libproxy.a แล้วทำการ Uncomment โดยลบเครื่องหมาย # ที่อยู่ข้างหน้าออก ถ้าไม่มีไฟล์ชื่อ Configuration ให้ทำสำเนาจากไฟล์ Configuration.tmpl ซึ่งมากับชุดโปรแกรม Apache ให้เป็นไฟล์ Configuration ก่อน สมมติว่าใช้โปรแกรม vi editor

Internet

```
#cd src/
#cp Contiguration.tmpl Configuration
#vi Configuration

...
## The proxy module enables the server to act as a proxy for outside
## http and ftp services. It's not as complete as it could be yet.
## NOTE: You do not want this module UNLESS you are running a proxy;
## it is not needed for normal (origin server) operation.

AddModule modules/proxy/libproxy.a
...
```

จากนั้นสั่งให้สคริปต์ "Contigure" ทำงาน โดยใช้คำสั่ง ./Configure ดังนี้

```
# ./Configure
Using config file: Configuration
 | Please note that the default httpd root directory has changed
 | from '/usr/local/etc/httpd/' to '/usr/local/apache/.'
 | You may add '-DHTTPD_ROOT=\"/usr/local/etc/httpd\"' to EXTRA_CFLAGS
 | in your Configuration file (and re-run Configure) or start
 | httpd with the option '-d /usr/local/etc/httpd' if you still
 | want to use the old root directory for your server.
Creating Makefile
 + configured for HP-UX platform
 + setting C compiler to gcc
 + setting C pre-processor to gcc -E
 + checking for system header files
 + adding selected modules
 + doing sanity check on compiler and options
Creating Makefile in support
Creating Makefile in main
Creating Makefile in ap
Creating Makefile in regex
Creating Makefile in os/unix
Creating Makefile in modules/standard
Creating Makefile in modules/proxy
#
```

สังเกตว่าบรรทัดล่างสุดจะมีการรวม modules/proxy ไว้ด้วย จากนั้นคอมไพล์โปรแกรมด้วยคำสั่ง make หากทุกอย่างถูกต้อง เราจะได้โปรแกรมชื่อ httpd อยู่ในไดเรคทอรี src/ มาใช้งาน

```
# make
...
===> ap
        rm -f libap.a
        ar cr libap.a ap_execve.o ap_cpystrn.o ap_signal.o  ap_slack.o ap_snprio
        /bin/true libap.a
<=== ap
===> main
        gcc  -DHPUX -DUSE_HSREGEX    -o gen_test_char gen_test_char.o  -lm
        ./gen_test_char >test_char.h
        gcc -c  -I../os/unix -I../include   -DHPUX -DUSE_HSREGEX  util.c
        rm -f libmain.a
        ar cr libmain.a alloc.o buff.o  http_config.o http_core.o http_log.o  ho
        /bin/true libmain.a
<=== main
===> modules
<=== modules
        gcc -c  -I./os/unix -I./include   -DHPUX -DUSE_HSREGEX  modules.c
        gcc -c  -I./os/unix -I./include   -DHPUX -DUSE_HSREGEX  buildmark.c
        gcc  -DHPUX -DUSE_HSREGEX     \
             -o httpd buildmark.o modules.o  modules/proxy/libproxy.a  modulesm
# ls -l httpd
-rwxrwxrwx   1 root     sys       659920 May 28 15:51 httpd
#
```

เมื่อได้ไฟล์ httpd ซึ่งเป็นโปรแกรมที่เราจะใช้งานมาแล้ว ก็ถึงขั้นตอนของการเซตค่าคอนฟิกูเรชันเพื่อกำหนดการทำงานต่าง ๆ ของเว็บเซิร์ฟเวอร์ ซึ่งในที่นี้เราจะกำหนดให้ทำหน้าที่เป็น Proxy ไฟล์ที่ใช้กำหนดคอนฟิกูเรชันของ httpd ที่มากับ Apache จะอยู่ในไดเรคทอรีย่อย conf/ ซึ่งจะมีอยู่ 3 ไฟล์ หนึ่งในไฟล์เหล่านั้นจะมีชื่อ httpd.conf-dist ให้ทำการสำเนาเป็น httpd.conf จากนั้นให้ใช้ Text Editor ทำการแก้ไขไฟล์นี้ให้กำหนดการทำงานของ httpd เป็น Proxy คำสั่งต่าง ๆ บางส่วนที่เกี่ยวข้องกับการทำงานเป็น Proxy แสดงให้เห็นในตารางที่ 1

สำหรับคำสั่งอื่น ๆ สามารถดูรายละเอียดการใช้งานได้จากเว็บไซท์ของ Apache ที่ http://www.apache.org คำสั่งที่สำคัญคือ ProxyRequests ถ้ากำหนดค่าเป็น on จะให้ทำหน้าที่เป็น Proxy หากเป็น off จะทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ธรรมดา สำหรับคำสั่งที่นำหน้าด้วย Cache จะใช้สำหรับกำหนดคุณสมบัติต่าง ๆ เกี่ยวกับแคช ลองดูตัวอย่างการเซตคอนฟิกูเรชันในไฟล์ httpd.conf เพื่อทำหน้าที่เป็น Proxy เซิร์ฟเวอร์ ต่อไปนี้

```
...
# Proxy Server directives. Uncomment the following lines to
# enable the proxy server:
#
<IfModule mod_proxy.c>
ProxyRequests On
#
#<Directory proxy:*>
#    Order deny,allow
#    Deny from all
#    Allow from .your_domain.com
#</Directory>
#
# Enable/disable the handling of HTTP/1.1 "Via:" headers.
# ("Full" adds the server version; "Block" removes all outgoing Via: headers)
# Set to one of: Off | On | Full | Block
#
#ProxyVia On

#
# To enable the cache as well, edit and uncomment the following lines:
# (no cacheing without CacheRoot)
#
CacheRoot "@@ServerRoot@@/proxy"
CacheSize 500000
CacheGcInterval 24
CacheMaxExpire 72
#CacheLastModifiedFactor 0.1
CacheDefaultExpire 72
#NoCache a_domain.com another_domain.edu joes.garage_sale.com

</IfModule>
# End of proxy directives.
...
```

Internet

Directive	ตัวอย่างการกำหนดค่า	ความหมาย
ProxyRequests	ProxyRequests on	กำหนดให้ทำงานเป็น Proxy
ProxyRemote	ProxyRemote * http://proxy.se-ed.com:8080	กำหนดให้ติดต่อผ่าน Proxy อื่นอีกทอดหนึ่ง
ProxyBlock	ProxyBlock forbid.site.com	กำหนดรายชื่อเว็บไซท์ที่ไม่ต้องการให้เข้าไปใช้งานได้ หากมีหลายไซท์ให้แยกกันด้วยช่องว่าง
NoProxy	ProxyRemote * http://proxy.se-ed.com:8000 NoProxy .se-ed.com 111.111.111.111	ใช้สำหรับการทำงานในอินทราเน็ต เมื่อต้องการให้ติดต่อกับเซิร์ฟเวอร์ปลายทางโดยตรงโดยไม่ผ่าน ProxyRemote
CacheRoot	CacheRoot /usr/local /apache/proxy/	ใช้กำหนดชื่อไดเรคทอรีที่ต้องการใช้เป็นแคช
CacheSize	CacheSize 20000	กำหนดขนาดของแคช มีหน่วยเป็น KByte
CacheGcInterval	CacheGcInterval 24	ใช้กำหนดเวลาในการตรวจสอบขนาดของ CacheSize เป็นชั่วโมง หากขนาดของข้อมูลในแคชเกินกว่าที่กำหนดใน CacheSize จะทำการลบส่วนที่เกินออก
CacheMaxExpire	CacheMaxExpire 168	ใช้กับเอกสาร HTTP สำหรับกำหนดเวลาสูงสุดที่จะเก็บข้อมูลนั้น ๆ ไว้ในแคชเป็นชั่วโมง หากเกินกว่าเวลานี้จะถือว่าหมดอายุ และจะไม่นำข้อมูลนั้นมาใช้งานอีก
CacheDefaultExpire	CacheDefaultExpire 24	คล้ายกับ CacheMaxExpire แต่ใช้กับโปรโตคอลที่ไม่มีการกำหนดเวลาหมดอายุเท่านั้น

ตารางที่ 1 คำสั่งสำคัญที่เกี่ยวข้องกับการสั่งให้ Apache ทำงานเป็น Proxy บางค่าที่จำเป็น

ตามตัวอย่างข้างต้นเป็นสั่งให้ Apache ทำงานเป็น Proxy โดยใช้ไดเรคทอรีย่อย proxy/ จาก ServerRoot เป็นพื้นที่สำหรับแคช มีขนาด 500 Mbyte ตรวจสอบขนาดของแคชนี้ทุก ๆ 24 ชั่วโมง ข้อมูลในแคชมีอายุเพียง 72 ชั่วโมง หรือ 3 วัน การกำหนดรายละเอียดต่าง ๆ เกี่ยวกับแคชนี้จะขึ้นอยู่กับปัจจัยหลายประการ คุณต้องพิจารณาให้เหมาะสมกับระบบของคุณเอง

ในกรณีที่ต้องการให้ Proxy ของเราใช้ข้อมูลโดยตรงจาก Proxy ตัวอื่น ในลักษณะเดียวกับรูปที่ 3 คือมีการใช้ Proxy ต่อกันเป็นทอด ๆ สามารถกำหนดได้จากคำสั่ง ProxyRemote ซึ่งมีรูปแบบเป็น

```
ProxyRemote <match> <remote-server>
```

โดยที่ <match> เป็นค่า URL ของเซิร์ฟเวอร์ที่ต้องการให้ผ่านทาง <remote-server> ได้ กรณีระบุเป็น * จะหมายถึงให้ใช้ข้อมูลทุกอย่างผ่านทาง <remote-server> ทั้งหมด รูปแบบของ <remote-server> อยู่ในรูปของ <protocol>://<hostname>[:port] เช่น

```
ProxyRemote * http://proxy.isp.co.th:8080
```

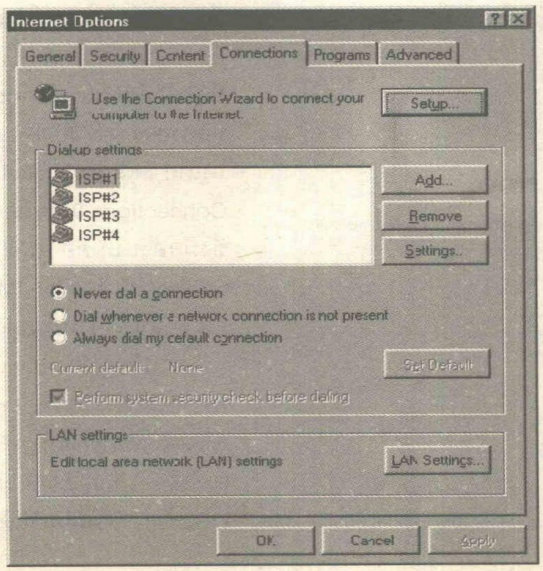
เป็นการกำหนดให้ Proxy ติดต่อขอข้อมูลจาก Proxy http://proxy.isp.co.th ทางพอร์ตหมายเลข 8080 คำสั่งนี้มีประโยชน์ในกรณีใช้งานในหน่วยงานที่มีจำนวนผู้ใช้งานมาก หรือหน่วยงานขนาดใหญ่ที่มีการใช้งานเครือข่ายคอมพิวเตอร์หลายเครือข่าย และเชื่อมโยงการใช้งานด้วยวงจรสื่อสารข้อมูลจำนวนมาก สามารถติดต่อเข้าสู่เครือข่ายอินเทอร์เน็ตผ่านทาง Proxy หลัก ส่วนเครือข่ายย่อย ๆ ก็อาจติดตั้ง Proxy ย่อยให้ติดต่อผ่านทาง Proxy หลักอีกทอดหนึ่ง ซึ่งจะสามารถประหยัดค่าใช้จ่ายในการเชื่อมโยงเครือข่ายเข้าสู่อินเทอร์เน็ตลงได้มาก เนื่องจากมีการใช้ข้อมูลร่วมกันมากขึ้น

หลังจากทำการแก้ไขไฟล์ httpd.conf แล้ว สามารถสั่งให้โปรแกรมทำงานโดยคำสั่ง

```
$ /usr/local/apache/httpd -f /usr/local/apache/conf/httpd.conf
```

ทั้งนี้สมมติว่าคุณติดตั้งโปรแกรมทั้งหมดไว้ในไดเรคทอรี /usr/local/apache และคอนฟิกูเรชันไฟล์ไว้ในไดเรคทอรี /usr/local/apache/conf/ หากทุกอย่างถูกต้อง คุณจะมีโปรแกรมที่ทำงานอยู่หลังฉากที่คอยรองรับการทำงานของผู้ใช้ที่ต้องการใช้งานผ่าน Proxy นี้ คุณสามารถตรวจสอบว่า Proxy ทำงานหรือไม่ได้โดยใช้คำสั่ง ps -ef | grep httpd

รูปที่ 6 ➲ ใน IE5 สามารถแยกการปรับตั้งค่าสำหรับการใช้ Proxy แยกระหว่าง LAN และ Dialup แต่ละแห่งได้อย่างอิสระ



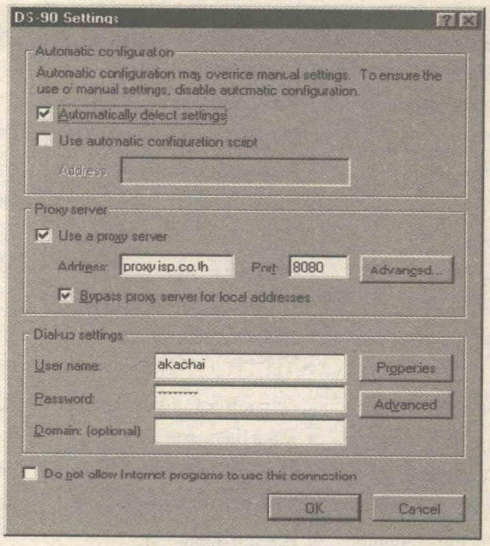
การติดตั้ง Apache เป็น Proxy บน Windows

Apache มีกำเนิดบนระบบปฏิบัติการ Unix แม้ปัจจุบันจะมีการแจกจ่ายรุ่นที่ทำงานบนระบบ Windows ด้วยก็ตาม แต่ยังคงถือว่าเป็นรุ่นทดสอบอยู่ อย่างไรก็ตาม Apache แจกจ่ายพร้อมกับซอร์สโค้ด โดยรุ่นที่ทำงานบนระบบ Windows ได้รับการคอมไพล์ด้วย Microsoft Visual C++ 5.0 หากคุณต้องการเพิ่มเติมบางสิ่งบางอย่าง คุณก็สามารถดำเนินการได้ด้วยตัวเอง สำหรับโปรแกรมที่แจกจ่ายสำหรับระบบปฏิบัติการ Windows นี้มีการคอมไพล์รวมส่วนที่ทำหน้าที่เป็น Proxy ไว้แล้ว และมีการใช้งานคล้ายกับในระบบ Unix คุณสามารถควบคุมหรือสั่งการให้โปรแกรมทำงานตามที่ต้องการได้โดยผ่านทางคอนฟิกูเรชันไฟล์ ปกติ Apache จะใช้อยู่ 3 ไฟล์คือ httpd.conf, access.conf และ src.conf แต่คุณสามารถสร้างขึ้นใหม่โดยใช้ชื่ออะไรก็ได้ การสั่งให้ Apache ทำงานเป็น Proxy ที่ง่ายที่สุดก็โดยการแก้ไขในไฟล์ httpd.conf ให้เพิ่มคำสั่งในการทำงานเป็น Proxy ไว้ด้วย คำสั่งต่าง ๆ จะคล้ายกับในระบบ Unix ให้ใช้ NotePad หรือ Text Editor อื่นก็ได้ ทำการแก้ไขค่า #LoadModule proxy_module modules /ApacheModuleProxy.dll โดยลบเครื่องหมาย # ข้างหน้าออกไป รวมถึงเครื่องหมาย # หน้าคำสั่ง ProxyRequest On ด้วย เพื่อให้โปรแกรมทำหน้าที่เป็น Proxy

```
...
LoadModule proxy_module modules/ApacheModuleProxy.dll
...
...
# Proxy Server directives. Uncomment the following line to
# enable the proxy server:

ProxyRequests On

# To enable the cache as well, edit and uncomment the following lines:
CacheRoot "C:/Program Files/Apache Group/Apache/proxy"
CacheSize 50000
CacheGcInterval 24
CacheMaxExpire 72
CacheLastModifiedFactor 0.1
CacheDefaultExpire 72
#NoCache a_domain.com another_domain.edu joes.garage_sale.com
...
```



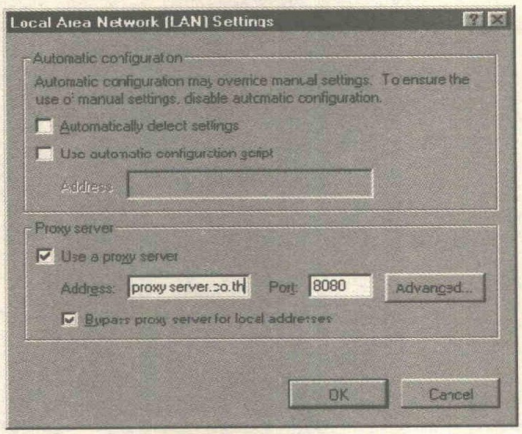
รูปที่ 7 การกำหนดให้ใช้งานผ่าน Proxy กรณีใช้งานผ่านทาง Dialup ใน IE5

กรณีที่ต้องการให้ใช้ข้อมูลจาก Proxy อื่นก็สามารถใช้คำสั่ง ProxyRemote เช่นเดียวกับกรณีของการใช้งานบน Unix โดยอาจใช้คำสั่งนี้ต่อท้ายคำสั่ง ProxyRequest ก็ได้ จากนั้นกำหนดค่าต่าง ๆ ที่เกี่ยวกับแคช หลังจากทำการแก้ไขแล้ว ให้ทำการบันทึก ในครั้งต่อไปที่สั่งให้โปรแกรม Apache ทำงาน มันก็จะทำหน้าที่เป็น Proxy/Cache Server

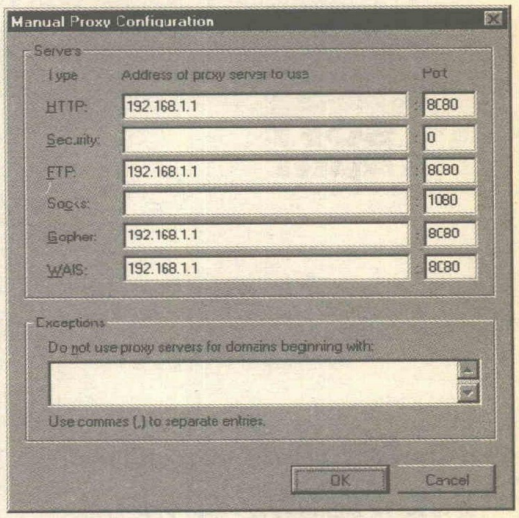
ทดสอบการทำงานของ Proxy

หลังจากสั่งงานให้ Apache ทำงานเป็น Proxy แล้ว ก็ถึงเวลาทดสอบว่า Proxy ของเราสามารถทำหน้าที่ตามที่ต้องการ โดยการใช้เครื่องที่อยู่บนเครือข่าย เรียกใช้เบราเซอร์ที่คุ้นเคย ปรับตั้งค่าให้ทำงานผ่านทาง Proxy ที่เพิ่งสั่งให้ทำงาน จากนั้นลองเรียกไปยังเว็บไซท์ต่าง ๆ ที่ต้องการ สำหรับ IE5 สามารถตั้งให้ทำงานผ่าน Proxy ได้โดย คลิกที่ Tools/Internet Options... เลือกแท็บ Connections แล้วเลือกวิธีการติดต่อเข้าสู่อินเทอร์เน็ต ซึ่งมีสองแบบคือ Dial Up Setting และผ่าน Lan Setting ให้เลือกวิธีการที่คุณใช้ในการเข้าสู่อินเทอร์เน็ต แล้วคลิกเครื่องหมายถูกที่ Proxy Server

สำหรับผู้ใช้ Netscape Navigator สามารถตั้งให้เบราเซอร์ทำงานผ่าน Proxy โดยคลิกที่ Edit/Preference แล้วคลิกที่เครื่องหมาย + หน้า Advanced แล้วคลิกที่ Proxies เลือก Manual Proxy Configuration ใส่ค่า URL ของเซิร์ฟเวอร์ที่ต้องการให้ทำหน้าที่เป็น Proxy



รูปที่ 8 การกำหนดให้ใช้งานผ่าน Proxy กรณีใช้งานผ่านทาง LAN ใน IE5



รูปที่ 9 การกำหนดให้ใช้งานผ่าน Proxy ใน Netscape Navigator 4.X

หลังจากนั้นก็ทดลองเรียกใช้เบราเซอร์ติดต่อไปยังเว็บไซท์ต่าง ๆ เบราเซอร์จะทำการติดต่อกับ Proxy ก่อนเสมอ จากนั้น Proxy จึงติดต่อกับเซิร์ฟเวอร์ปลายทางหรือค้นหาข้อมูลที่มีอยู่ในแคช ส่งกลับมายังเบราเซอร์ เราสามารถสังเกตการติดต่อของเบราเซอร์ได้จากแถบแสดงการทำงานด้านล่าง หากทุกอย่างถูกต้องก็จะสามารถใช้งานได้เช่นเดียวกับการท่องอินเทอร์เน็ตตามปกติ โดยเราสามารถทำการเซตเครื่อง Workstation บนระบบเครือข่ายให้ใช้งานผ่าน Proxy ที่ติดตั้งไว้ได้ทุกเครื่อง

ข้อจำกัดของการใช้งาน Proxy ผ่านโปรแกรม Apache ก็คือ เราสามารถใช้งานได้เฉพาะโปรโตคอล HTTP และ FTP หากคอมพิวเตอร์ที่ใช้งานไม่ได้ต่อเข้ากับอินเทอร์เน็ตโดยตรง ก็จะไม่สามารถใช้อีเมล์หรือ ICQ ได้ ยกเว้นจะติดต่อโดยวิธีอื่น

บทสรุป

Proxy/Cache Server ช่วยให้การใช้งานวงจรเครือข่ายเป็นไปอย่างมีประสิทธิภาพ ข้อมูลใดที่เคยมีการดาวน์โหลดมาแล้วก็ไม่จำเป็นจะต้องเสียเวลา เสียวงจรไปดาวน์โหลดมาอีก ทำให้เราสามารถประหยัดค่าเช่าวงจรเครือข่ายลงได้ โดยจ่ายค่าเช่าวงจรเท่าที่จำเป็นเท่านั้น เป็นที่ทราบกันดีว่าค่าเช่าวงจรสื่อสารเป็นค่าใช้จ่ายที่ค่อนข้างสูงและต้องจ่ายอย่างต่อเนื่อง แล้วคุณจะไม่นำ Proxy มาใช้ประโยชน์บ้างหรือ